

第一章

1. 操作系统是计算机中管理硬件和软件资源的系统软件
2. 实时操作系统是能优先处理紧急任务的操作系统
3. 互斥共享是某个资源一个时间段只允许一个进程访问该资源

三.

1. 硬件 软件
3. 硬件资源 软件资源
5. 共享性 异步性

第二章

3. 临界区是访问共享资源那段代码
4. 进程同步是多道程序下, 对并发执行进程协调, 使按照一定顺序访问共享资源

三.

1. 动态性 异步性 结构性
5. 动态 静态
6. 竞争
8. 共享内存 消息传递 共享文件
9. 就绪 运行 阻塞 终止

五.

- main() {
2. Semaphore a, b, c, d, e, f, g = 0, 0, 0, 0, 0, 0, 0
- cobegin
- { S1; signal(a); signal(b); }

记录。为了进一步提高检索效率，可以为顺序文件建立多组索引，即多索引。

No.

Date

```

{ wait (a); S2; signal (a); signal (d); }
{ wait (b); S3; signal (e); }
{ wait (c); S4; signal (f); }
{ wait (d); S5; signal (g); }
{ wait (e); wait (f); wait (g); S6 }
corend }

```

cobegin

3.

cobegin

Semaphore mutex = 1, empty = 5, apple = 0, orange = 0;

Father()	Son()	daughter()
{ while (1)	{ while (1)	{ while (1)
{ P(empty);	{ P(orange);	{ P(apple);
P(mutex);	P(mutex);	P(mutex);
if (苹果) {	V(empty);	V(empty);
{ V(apple); }	V(mutex); }	V(mutex); }
else { V(orange); }		
V(mutex); }		

corend

4.

cobegin

Semaphore mutex1, mutex2, empty1, empty2 = 1, full1, full2 = 0

No. _____
Date _____

```

void PA()
{ while(1)
  { P(Cempty1);
    P(Cmutex1);
    ...
    V(Cfull1);
    V(Cmutex1);
  }
}

void PB()
{ while(1)
  { P(Cfull1);
    P(Cmutex1);
    ...
    V(Cempty1);
    V(Cmutex1);
    P(Cempty2);
    P(Cmutex2);
    ...
    V(Cfull2);
    V(Cmutex2);
  }
}

void PC()
{ while(1)
  { P(Cfull2);
    P(Cmutex2);
    ...
    V(Cempty2);
    V(Cmutex2);
  }
}
  
```

6. cobegin

semaphore customers, barber, payment, receipt = 0 mutex = 1

int waiting = 0, waiting-room = N

```

void barber()
{ while(1)
  { P(customers);
    P(mutex);
    waiting--;
    V(barber);
    V(mutex);
    P(payment);
    V(receipt);
  }
}

void customer()
{ while(1)
  { P(mutex);
    if(waiting > waiting-room)
    { V(mutex);
      waiting++;
      V(customers);
      V(mutex);
      P(barber);
      V(payment);
      P(receipt);
    }
  }
}
  
```


② SJF (抢占)

进程	到达	服务	完成	周转	带权
A	0	3	3	3	1
B	2	6	20	18	3
C	4	4	8	4	1
D	6	5	15	11	2.2
E	8	2	10	2	1

$\text{平均带权} = \frac{1+3+1+2.2+1}{5} = 1.64$
 $\text{平均周转} = \frac{3+18+8+11+2}{5} = 7.2$

④ HRRN

进程	到达	服务	完成	周转	带权
A	0	3	3	3	1
B	2	6	9	7	1.6
C	4	4	13	9	2.25
D	6	5	20	14	2.8
E	8	2	15	7	3.5

$D: \frac{14}{5} = 2.8$
 $E: \frac{7}{2} = 3.5$
 $C: \frac{9}{4} = 2.25$
 $B: \frac{7}{4} = 1.75$

⑤ RR

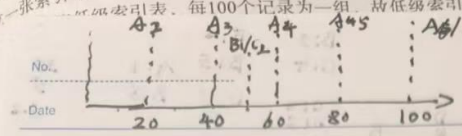
进程	到达	服务	完成	周转	带权
A	0	3	4	4	1.33
B	2	6	18	16	8
C	4	4	17	13	1.75
D	6	5	20	14	2.8
E	8	2	15	7	3.5

$\text{平均周转} = 9.6$
 $\text{平均带权} = 3.18$

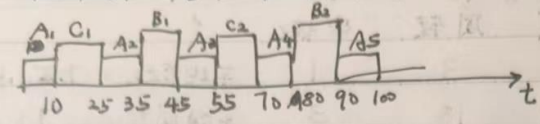
2. A

X

1000个记录。为了进一步提高检索效率，可以为顺序文件建立多级索引，即为索引顺序索引表，从而形成两级索引表。例如，对于一个含有10⁶个记录的顺序文件，每100个记录为一组，故低级索引表应含有10⁴个表项，在



2. 由于 $\frac{10}{20} + \frac{10}{40} + \frac{10}{60} = 1$ 可以接纳



$B_1: 50 - 25 - 10 = 15$ 70 80-10
 $A_2: 40 - 25 - 10 = 5$
 $A_3: 5$ $B_2: 10$ $A_5: 10$

3. 进程	Max	Need	Allocation	Available
$\checkmark P_0$	0044	0012	0032	
$\checkmark P_1$	2750	1750	1000	1622
P_2	361010	2356	1354	004000
$\checkmark P_3$	0984	0652	0332	
P_4	06610	0656	0014	

进程	Work	Need	Allocation	W+A	Finish
P_0	520 622	0012	0032	1654	T
P_3	1654	0652	1354	29108	T
P_1	29108	1750	1000	39108	T
P_2	39108	2356	1354	4121512	T
P_4	4121512	0656	0014	4121616	T

存在安全序列 { P_0, P_3, P_1, P_2, P_4 }

2) Request \leq Need Request $<$ Available 进行假设

进程	Work	Need	Allocation	W+A	Finish
	0400				

不存在安全序列

3) 不一定